Name: Dang Khoa Le

Student ID: 103844421

**TNE30024**

Deploying Secure Engineering Applications Online

# Tutorial 1 Symmetric Key Cryptography

**Questions**

1. Use the one-time-pad 1010011000 to encrypt and decrypt the message 1111011001

**Encryption:** To encrypt the message 1111011001 using the one-time pad 1010011000, perform a bitwise XOR operation:

* Message: 1111011001
* One-Time Pad: 1010011000

1 XOR 1 = 0

1 XOR 0 = 1

1 XOR 1 = 0

1 XOR 0 = 1

0 XOR 1 = 1

1 XOR 0 = 1

1 XOR 1 = 0

0 XOR 1 = 1

1 XOR 0 = 1

0 XOR 0 = 0

0 XOR 0 = 0

1 XOR 1 = 0

0 XOR 0 = 0

1 XOR 0 = 1

1 XOR 1 = 0

So the encrypted message is 0101110010.

**Decryption:** To decrypt 1111011001 using the same one-time pad 1010011000, perform a bitwise XOR operation:

0 XOR 1 = 1

1 XOR 0 = 1

0 XOR 1 = 1

1 XOR 0 = 1

1 XOR 1 = 0

1 XOR 0 = 1

0 XOR 1 = 1

1 XOR 1 = 0

1 XOR 0 = 1

0 XOR 0 = 0

0 XOR 0 = 0

0 XOR 1 = 1

1 XOR 0 = 1

0 XOR 1 = 1

0 XOR 1 = 1

So the decrypted message is 1111011001, which matches the original message.

1. The following S-Box (S1 from the DES standard) maps a six bit input to a four bit output. What will be the output of this box when presented with an input of 7. (All values are base 10.)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Row /  Column | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| 1 | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| 2 | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| 3 | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

To use this S-Box, we need to convert the decimal input to binary, select the row and column, and find the output.

1. Convert the decimal input 7 to a 6-bit binary number: 000111
2. Determine the row and column from the binary number:
   * Row: Formed by the first and last bits of the 6-bit binary number 000111, which is 01 in binary or 1 in decimal.
   * Column: Formed by the middle four bits of the 6-bit binary number 000111, which is 0011 in binary or 3 in decimal.
3. Look up the value in the S-Box table at Row 1, Column 3:
   * The value at Row 1, Column 3 is 4.

So, the output for the input 7 is **4**.

1. Consider the following simplified block encryption scheme:

Plaintext is encrypted a byte at a time using the following steps:

* + Step 1.
    - The plain text is expanded to 12 bits by duplicating the first and last two bits (ie, abcdefgh becomes aabbcdefgghh
  + Step 2. o A 12 bit sub key is XORed with the expanded text from step 1
  + Step 3.
    - The bit sequence from step 2 is split into two 6 bit sequences and fed into the following two S-BOXes
  + Step 4.
    - The output of the S-BOXes is concatenated and fed through a permutation process that reverses the bit sequence order

What is the output for a plaintext input of 1001 0100 and a 12 bit sub-key of 1001 0011 1010?

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S1 | | **Middle four bits** | | | | | | | | | | | | | | | |
| **0000** | **0001** | **0010** | **0011** | **0100** | **0101** | **0110** | **0111** | **1000** | **1001** | **1010** | **1011** | **1100** | **1101** | **1110** | **1111** |
| Outer  bits | **00** | 00 | 0010 | 1100 | 0100 | 0001 | 0111 | 1010 | 1011 | 0110 | 1000 | 0101 | 0011 | 1111 | 1101 | 0000 | 1110 |
| **01** | 01 | 1110 | 1011 | 0010 | 1100 | 0100 | 0111 | 1101 | 0001 | 0101 | 0000 | 1111 | 1010 | 0011 | 1001 | 1000 |
| **10** | 10 | 0100 | 0010 | 0001 | 1011 | 1010 | 1101 | 0111 | 1000 | 1111 | 1001 | 1100 | 0101 | 0110 | 0011 | 0000 |
| **11** | 11 | 1011 | 1000 | 1100 | 0111 | 0001 | 1110 | 0010 | 1101 | 0110 | 1111 | 0000 | 1001 | 1010 | 0100 | 0101 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S2 | | **Middle four bits** | | | | | | | | | | | | | | | |
| **0000** | **0001** | **0010** | **0011** | **0100** | **0101** | **0110** | **0111** | **1000** | **1001** | **1010** | **1011** | **1100** | **1101** | **1110** | **1111** |
| Outer  bits | **00** | 00 | 0010 | 1100 | 0100 | 0001 | 0111 | 1010 | 1011 | 0110 | 1000 | 0101 | 0011 | 1111 | 1101 | 0000 | 1110 |
| **01** | 01 | 1110 | 1011 | 0010 | 1100 | 0100 | 0111 | 1101 | 0001 | 0101 | 0000 | 1111 | 1010 | 0011 | 1001 | 1000 |
| **10** | 10 | 0100 | 0010 | 0001 | 1011 | 1010 | 1101 | 0111 | 1000 | 1111 | 1001 | 1100 | 0101 | 0110 | 0011 | 0000 |
| **11** | 11 | 1011 | 1000 | 1100 | 0111 | 0001 | 1110 | 0010 | 1101 | 0110 | 1111 | 0000 | 1001 | 1010 | 0100 | 0101 |

1. **Plain Text Expansion**:
   * Plaintext input: 1001 0100
   * Expand it to 12 bits by duplicating the first and last two bits: 1010 1001 0100 0100
2. **XOR with Sub-Key**:
   * 12-bit sub-key: 1001 0011 1010
   * XOR operation:

yaml

Sao chép mã

1010 1001 0100 0100

XOR 1001 0011 1010

-------------------

0011 1010 1110

1. **Split into Two 6-bit Sequences**:
   * The resulting 12-bit sequence from XOR operation: 0011 1010 1110
   * Split into two 6-bit sequences: 001110 and 101110
2. **S-Box Substitution**:
   * **First S-Box (S1)**:
     + Input: 001110
     + Row: First and last bits 00, which is 0 in decimal
     + Column: Middle four bits 0111, which is 7 in decimal
     + Value from S1 table at Row 0, Column 7: 8
   * **Second S-Box (S2)**:
     + Input: 101110
     + Row: First and last bits 10, which is 2 in decimal
     + Column: Middle four bits 0111, which is 7 in decimal
     + Value from S2 table at Row 2, Column 7: 8
3. **Concatenate S-Box Outputs and Permutation**:
   * S-Box outputs: 1000 from S1 and 1000 from S2
   * Concatenate: 10001000
   * Reverse the bit sequence: 00011001

So, the final output for a plaintext input of 1001 0100 and a 12-bit sub-key of 1001 0011 1010 is **00011001**.

# Public Key Cryptography

**Questions**

1. Bob wishes to send a message to Alice. He wants to encrypt it and digitally sign it using public key encryption.
   1. Which key will Bob use to encrypt the message?

Bob will use Alice's **public key** to encrypt the message.

* 1. Which key will Bob use to sign the message?

Bob will use his **private key** to sign the message.

* 1. Which key will Alice use to decrypt the message?

Alice will use her **private key** to decrypt the message.

* 1. Which key will Alice use to validate the digital signature?

Alice will use Bob's **public key** to validate the digital signature.

1. Is the Diffie-Hellman algorithm a public key encryption algorithm? If not, what is it?

The Diffie-Hellman algorithm is not a public key encryption algorithm; it is a key exchange algorithm. It allows two parties to securely share a secret key over an insecure channel.

1. 133 is the product of two primes. What are they?

To factorize 133, try dividing by prime numbers:

* 133 ÷ 7 ≈ 19 (no remainder)

So, 133 = 7 × 19.

1. RSA and Diffie-Hellman can generate very large numbers that require their modulus to be calculated. Fortunately, modulo arithmetic is associative and commutative. That is:

ap+q+r mod N = (ap mod N) (aq mod N) (ar mod N) mod N

For example

36 mod 5 = (32 mod 5) (32 mod 5) (32 mod 5) mod 5

= (9 mod 5)(9 mod 5)(9 mod 5) mod 5

= 43 mod 5 = 64 mod 5 = 4

Try this approach with 55 mod 23

For 55 mod 23 using the formula:

55 mod 23 = (55 mod 23)

= (55 - 2\*23)

= 55 - 46

= 9

Thus, 55 mod 23 is 9.

1. What key do Alice and Bob come to agree upon using the Diffie-Hellman algorithm using the following values?

Alice chooses a = 3, Bob chooses b = 4, p = 17 and g = 3.

Using the Diffie-Hellman algorithm with given values:

* Alice’s value: a=3
* Bob’s value: b=b = 4b=4
* Prime p=17
* Generator g=3

Alice computes: A = g^a mod p = 3^3 mod 17 = 27 mod 17 = 10

Bob computes: B= g^b mod p = 3^4 mod 17 = 81 mod 17 = 13

Both Alice and Bob then compute the shared key:

* Alice computes: K = B^a mod p = 13^3 mod 17 = 2197 mod 17 = 7
* Bob computes: K = A^b mod p = 10^4 mod 17 = 10000 mod 17 = 7

So, the shared key is 7.

1. The following is a public/private key pair.

[3,33] and [7,33]

Use the keys and RSA to encrypt and decrypt‘2’.

Given public/private key pair: [3,33] and [7,33].

Encrypting 2:

Ciphertext \( C = 2^3 \mod 33 = 8 \mod 33 = 8 \)

Decrypting 8:

Plaintext \( M = 8^7 \mod 33 = 2097152 \mod 33 = 2 \)

So, encryption and decryption are consistent.

1. Generate a public / private key using the prime numbers 3 and 11.

Using primes 3 and 11:

1. Compute n=p×q=3×11=33.
2. Compute ϕ(n)=(p−1)×(q−1)=2×10=20.
3. Choose a public key exponent e such that 1<e<ϕ(n) and gcd(e,ϕ(n)e= 1. Here, e=3 is a valid choice.
4. Compute the private key exponent d such that e×d = 1mod ϕ(n). For e=3e, d=7.

Thus, the key pair is: 3,33] (public) and [7,33] (private).